

Formal Specification of Holonic Control System ADACOR Product Holon, using High-Level Petri Nets

Paulo Leitão¹, Armando W. Colombo², Francisco Restivo³, Ronald Schoop⁴

¹ Polytechnic Institute of Bragança
Quinta S^{ua} Apolónia, Apartado 134, 5301-857 Bragança, Portugal
E-mail: pleitao@ipb.pt

² Schneider Electric GmbH
Steinheimer Str. 117, 63500 Seligenstadt, Germany
E-mail: armando.colombo@modicon.com

³ Faculty of Engineering, University of Porto
Rua Dr. Roberto Frias, P-4200-465 Porto, Portugal
E-mail: fjr@fe.up.pt

⁴ Schneider Automation SA
245 route des Lucioles BP147, Sophia Antipolis, France
E-mail: ronald.schoop@modicon.com

Abstract – Holonic manufacturing and multi-agent paradigms are suitable to support the actual challenges of flexible manufacturing systems, due to their decentralisation, modularity and autonomy features. The formal specification assumes a critical role in order to understand and synthesise those complex systems. The Petri Nets formalism is adequate to model and validate the dynamic behaviour, but present weak points when the system contains many instances of the same component, since the model grows (structure and components) in a non-controllable manner. The use of High-Level Petri Nets, allows to reduce this complexity, by compressing the representation of states, actions and events, to overcome the identified limitations and to support more complex and bigger coordination scenarios. This paper presents a formal specification of the ADACOR product holons using High-Level Petri Nets and the associated formal validation of the model.

Keywords: Flexible Manufacturing Systems, Holonic Manufacturing Systems, Coloured Petri Nets, Formal Specification, Modelling.

I. INTRODUCTION

The flexible manufacturing systems are complex and stochastic environments requiring the development of flexible, agile and intelligent management and control architectures that support small batches, product diversity, high quality and low costs, imposed by global markets. Holonic and multi-agent approaches seems to be suitable to face these requirements due to their decentralisation, modularity, autonomy and re-use control software features.

In the last years, several architectures and developments in holonic manufacturing were reported, covering different

application domains, such as manufacturing scheduling and control, materials handling, machine controllers and assembly systems, as referred in [1-5], and others compiled in [6].

One of these holonic architectures, proposed during the last two years, is the ADACOR (Adaptive Holonic Control Architecture for Distributed Manufacturing Systems) architecture [7], which aims to improve the performance of control system in industrial stochastic scenarios, characterised by the frequent occurrence of unexpected disturbances.

ADACOR is based in a set of autonomous, cooperative, self-organised and intelligent holons, grouped into four main holon classes, Fig. 1: product, task, operational and supervisor holons [7].

Each available product is represented by a product holon that contains all knowledge related to the product and process. Production orders to be executed in the factory plant are represented by task holons, which are responsible for the control and supervision of their execution. The operational holons represent the manufacturing resources, such as operators and robots, managing its behaviour and agenda according the resource goals, constraints and skills. The product, task and operational holons are quite similar to the product, order and resource holons, presented at the PROSA reference architecture [1]. The supervision holon, presents different characteristics than the staff holons defined in PROSA, introducing coordination and global optimisation in decentralised control approaches, coordinating several operational and supervisor holons. In normal operation, the supervisor holon supervises and regulates the activity of the holons under its domain, while when a disturbance occurs, these holons may have to find their way without the help of

the supervisor holon. The supervisor holon is also responsible for the group formation, based in pre-defined clusters of holons, combining synergies, aggregating skills and offering the combined services to external entities in the manufacturing system. These groups can be formed to build a shop floor, a manufacturing cell, or a machine equipped with a set of tools, assuming the supervisor holon the control role of each group.

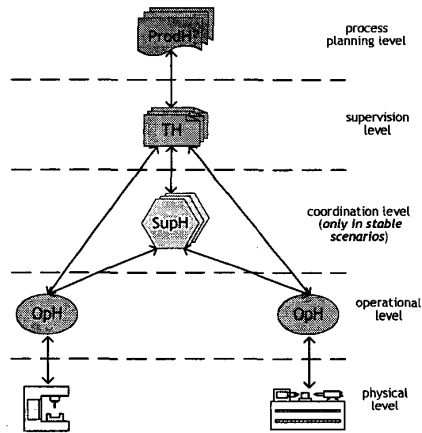


Fig. 1 Interdependency between ADACOR Holon Classes

In order to formalize the structure and the behaviour of the holonic manufacturing control systems, and to validate its behaviours and particularly to analyse the co-operation and interaction between the distributed holons, aiming to understand and synthesize the structure and behaviour of the system, it is important to count with a formal modelling methodology.

The modelling of the dynamic behaviour of the system requires a formal tool that captures characteristics like concurrency or parallelism, asynchronous operations, deadlocks, conflicts solutions and resource sharing, which are inherent to ADACOR. Additionally, it is crucial that the formal modelling tool has the capability to validate the behavioural characteristics of these event-driven systems, as also the analysis of other important aspects, such as the deadlock detection and the performance analysis.

The methodology proposed in [8] for the formal modelling of holonic applications, as illustrated in Fig. 2, combines the UML (Unified Modelling Language) and the PN modelling tools. UML is an object oriented based modelling tool that is adequate to model the structure and the static aspects of a manufacturing system. In the proposed formal methodology, the static aspects are modelled using mainly the class diagrams, which show the classes of objects in the system, the attributes and methods for each class, and the relationships between the objects.

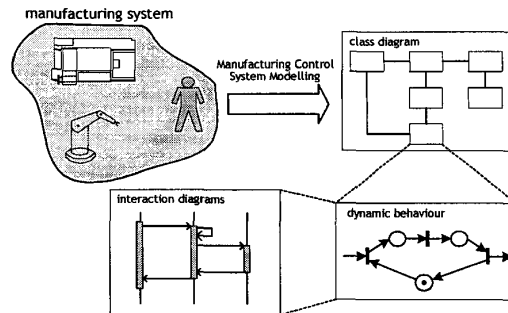


Fig. 2 Modelling a Manufacturing Control System

The UML modelling tool doesn't support efficiently the modelling of the dynamic behaviour aspects and the formal validation of these specifications. On the other hand, the PNs is a formal modelling tool, both graphical and mathematical that seems adequate to model and analyse the structure and the dynamic behaviour of complex event-driven systems with high distribution degree. In comparison with UML, the PN formalism allows designing the control system behaviour, but also to formally validate and verify the behaviour of the system, based in a strong mathematical background inherently embedded in the PN formalism. In this sense, the proposed methodology uses the PN formalism to model the dynamic behaviour of the holonic manufacturing control system. More details about PN theory and mathematical fundamentals are out of the scope of this work, but can be found in the following references [9, 10].

The formal methodology described in [8] to model the behaviour of the ADACOR-holon classes in a bottom-up approach, uses a kind of Petri net tailored for production management and control modelling purposes. The individual model of each holon uses special temporised transitions to model activities execution, that can be exploded into a more detailed and refined level. These sub-models, according to the degree of refinement, are the different software control modules of the hardware, i.e., a formal representation of the holons. The edition, simulation, analysis and formal validation of the structural and behavioural specifications of the ADACOR-holons can be found in [11].

However, in industrial manufacturing applications, the PN models become highly complex and difficult to handle and the developed models for the holon classes present some limitations. The modelling of simultaneous execution of different instances, such as the modelling of different resources using the same operational holon model or different product instances using the same product holon model, is a complex task that cannot be covered by the PN-formalism proposed in [8].

The use of High-level PN, such as those proposed in [12], allows reducing this complexity, by compressing the representation of states, actions and events, to overcome the identified limitations and to support more complex and bigger

coordination scenarios. In this case, for example each product has a colour-tone, allowing to have the same model as many times as the number of colour-tones.

This paper is organised as follows: First, Section 2 presents the formal notation and Section 3 the basic concepts and formal definitions associated to the Coloured Petri Nets. In Section 4 it is described the CPN model of one ADACOR holon, the product holon, that allows to model the simultaneous execution of different instances, supporting more complex and bigger coordination scenarios. Section 4 discusses the correctness of structure and behaviour of the model. Finally, Section 5 rounds up the paper with conclusions.

II. NOTATION

The notation that will be used in the formal modelling using High Level Petri Nets, will be briefly presented.

Consider N as the set of non-negative integers and Z as the set of integers. Let Ω be a finite set. The set of functions from Ω to N is denoted $\text{Bag}(\Omega)$.

An item a of $\text{Bag}(\Omega)$ is denoted $\sum_{u \in \Omega} a_u \cdot u$ where the summation is over $u \in \Omega$. It is also called a multi-set a .

A partial order on $\text{Bag}(\Omega)$ is defined by:

$$a \leq b \text{ if and only if } \forall u \in \Omega, a_u \leq b_u$$

The sum of two items of $\text{Bag}(\Omega)$ is defined by:

$$[a+b] = \sum (a_u + b_u) \cdot u, \forall u \in \Omega$$

The inner product of two items of $\text{Bag}(\Omega)$ is defined by:

$$[a \cdot b] = \sum (a_u \cdot b_u), \forall u \in \Omega$$

Then $\langle u_i, u_j \rangle = 0$ if $i \neq j$ and $\langle u_i, u_i \rangle = 1$ if $i = j$

$a_u = a \cdot \langle u, u \rangle$ show the multiplicity of the element $u \in \Omega$.

Let A and B two finite non-empty sets, then a function $f \in [A \rightarrow \text{Bag}(B)]$, where a and $\text{Bag}(B)$ are non-empty sets, can be uniquely extended to a linear function $f_{\text{ext}} \in [\text{Bag}(A) \rightarrow \text{Bag}(B)]$ called the multi-set,

$$f_{\text{ext}}: \forall c \in \text{Bag}(A): f_{\text{ext}}(c) = \sum c(a) \cdot f(a), \forall a \in A$$

As an example, consider the sets $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3, b_4\}$. Then we obtain:

- 1) multi-set $ms_1 = a_1 + 2 \cdot a_2 + a_3$ and multi-set $ms_2 = a_2 + 4 \cdot a_3$ are both members of $\text{Bag}(A)$.

- 2) if

$$f \in [A \rightarrow \text{Bag}(B)] \text{ is defined by } f(a_1) = b_1 + 2 \cdot b_3; \\ f(a_2) = b_2 + b_3 + b_4; f(a_3) = b_3 + 2 \cdot b_4$$

then $\forall c \in \text{Bag}(A)$,

$$\text{fext}(c) = c(a_1) \cdot (b_1 + 2 \cdot b_3) + c(a_2) \cdot (b_2 + b_3 + b_4) + \\ c(a_3) \cdot (b_3 + 2 \cdot b_4) = c(a_1) \cdot b_1 + c(a_2) \cdot b_2 + [2 \cdot c(a_1) + c(a_2)] \cdot b_3 + \\ c(a_3) \cdot b_4 + [c(a_2) + 2 \cdot c(a_3)] \cdot b_4$$

Remarks: In this work we use the terms function „ f “ and its linear extension function „ fext “ in the same way.

III. COLOURED PETRI NETS

A. Introduction

Coloured Petri Nets (CPN) are mathematical-graphical oriented formalisms for design, specification, validation and verification of concurrent systems [13,14]. A CPN is particularly well suited for systems where communication, synchronization, concurrency and competition on shared resources are important relationships. Typical examples of application areas are communication protocols, distributed systems, embedded systems, flexible manufacturing systems and workflow analysis. Without loss of rigor, the rest of the paper concentrates on their application in the holonic manufacturing area.

CPN have got their name because they allow the use of tokens that carry data values and can be distinguished from each other, in contrast to the tokens of low-level Petri nets, which by convention are drawn as black-dots.

B. Definitions

Let's recall here, under the definitions of a CPN, the firing rule of transitions, the incidence matrix and the net flows.

Definition 1: A CPN is a 7-tuple, [15],

$$\text{CPN} = \langle P, T, C, I^+, I^-, G, M_0 \rangle$$

satisfying the following requirements:

- $P = \{p_1, p_2, \dots, p_m\}$ is a finite set of places.
- $T = \{t_1, t_2, \dots, t_n\}$ is a finite set of transitions
- C is the colour function defined from $P \cup T$ into Ω ; Ω is a finite and not empty set. $C(s)$ is called the colour set of „ s “, and an item of $C(s)$ is called a colour of „ s “. C attaches to each place a set of possible token-colours $C(p)$ and to each transition a set of possible occurrence-colours or firing modes $C(t)$.
- $I^+ (I^-)$ are respectively the input function and the output function defined on $P \times T$, such that $I^+(p, t)$ is a function from $C(t) \times C(p)$ to N (i.e., a function from $C(t)$ to $\text{Bag}(C(p))$), $\forall (p, t) \in P \times T$.
- G is the *guard function*, defined from T into expressions of type Boolean, (i.e., a predicate), such that

$$\bullet \forall t \in T:$$

$$[\text{Type}(G(t)) = \text{Boolean} \wedge \text{Type}(\text{Variable}(G(t))) \subseteq C].$$

$$\bullet \forall t \in T \wedge \forall c_{ti}, c_{ij}, c_{ik} \in C(t) \Rightarrow$$

$$\Rightarrow G\&j(t) = (c_{ti} \wedge c_{ij} (\neg c_{ik}) \wedge \dots), i \neq j \neq k.$$

Each $G\&j(t)$ is a Boolean function of occurrence modes, related to the transition t , where c_{ik} is a Boolean variable $\Rightarrow (\neg c_{ik}) = 1$ exactly when $c_{ik} = 0$.

According to this definition, a standard form of a Guard function is described below:

$$G(t) = G\&1(t) \vee G\&2(t) \vee G\&3(t) \vee \dots$$

- M_0 is the initial marking of the net. It is a function defined on P , where $M_0(p)$ is a function from $C(p)$ to N (i.e., an item of $\text{Bag}(C(p))$), $\forall p \in P$.

The $M(p)$ and $M_0(p)$ markings give respectively the number of coloured tokens in the place „ p “ for the current and the initial marking.

Definition 2: The firing rule for CPNs is defined by:

- A transition t is enabled for a marking M and for a colour mode $c_t \in C(t)$ if and only if $\forall p \in {}^*t, M(p) \geq I(p, t, c_t)$.
- The firing of transition t for a marking M and a colour mode $c_t \in C(t)$ gives a new marking M' defined $\forall p \in P$ by,

$$M'(p) = M(p) - I^-(p, t, c_t) + I^+(p, t, c_t) \quad (1)$$

Definition 3: The incidence matrix I of a CPN is defined by

$$I(p, t, c_t) = I^-(p, t, c_t) - I^+(p, t, c_t)$$

Then I is a function from $C(p) \times C(t)$ to Z . $I(p, t, c_t)$ can also be viewed like a matrix of $\Omega(p, c) \times \Omega(t, c')$ of integers, where the first union is over $p \in P$ and $c \in C(p)$ and the second union is over $t \in T$ and $c' \in C(t)$, by the simple transformation $I((p, c), (t, c')) = I(p, t)(c, c')$.

Definition 4: A positive weighted set of places is a function S defined on P , such that

$$S(p): C(p) \rightarrow N$$

The operator „ \cdot “ is the generalization of matrix multiplication, by substituting each product for a function composition.

Definition 5: Let be the equations system

$$X^T \cdot I = 0$$

If $x = v$ is a solution and if equation (1) is multiplied on the left by v^T , for all reachable markings M and M' , then $v^T \cdot M = v^T \cdot M + v^T \cdot I \cdot D$, where D is a vector of the positive weighted set of transitions (firing modes of transitions) with dimension Cardinal $\text{Card}(T)$.

$$v^T \cdot M' = v^T \cdot M \text{ because } v^T \cdot I = 0 \quad (2)$$

The set of place-flows (p-flows) E of a CPN is defined by

$$E = \{v: \text{vector of coloured places} / v^T \cdot I = 0\}$$

The equation (2) is called a linear invariant of markings (place-invariants).

Definition 6: Let be the equations system

$$I \cdot X^T = 0$$

If $x = w$ is a solution and if equation (1) is multiplied on the right by w^T , for all reachable markings M and M' , then

$$M' \cdot w^T = M \cdot w^T + I \cdot w^T$$

where w^T is a vector of the positive weighted set of transitions (firing modes of transitions) with dimension $\text{Card}(T)$.

$$M' \cdot w^T = M \cdot w^T \text{ because } I \cdot w^T = 0, \text{ and } M' = M \quad (3)$$

The set of transition-flows (t-flows) F of a CPN is defined by,

$$F = \{w: \text{vector of firing modes of transitions} / I \cdot w^T = 0\}$$

The equation (3) is called a linear invariant of firings (transition-invariants).

In practice, there are some kind of colour domains and colour functions, which are frequently used. These are called standard colour domains and standard functions (for more information, the reader should consult [15]).

Definition 7: A set Ω_s is called “basic (standard) colour domain” and its elements “colour tones”. Ω_s can be extended to the ring $(\Omega_s, \oplus, \circ)$, where the arithmetic functions \oplus and \circ are executed module s (s is the cardinality of Ω_s) [15].

For example, a standard colour domain in this work can be defined as $PH = (ph_1, ph_2, \dots, ph_n)$. PH is the colour product holon and each element is a product holon (colour tone ph_i).

Note: The basic colour domain $\Omega_k = \{\langle \bullet \rangle\}$ is the set, which unique element is the discoloured token corresponding to the classical definition of marking [9].

Definition 8: A complex colour domain is defined as the Cartesian product of two or more basic colour domains.

Definition 9: The Universal colour domain Ω^* of the net will be the Cartesian product of all basic colour domains. That is $\Omega^* = \prod_{j \in [1:n]} \Omega_j$ and $\forall \omega^* \in \Omega^* \Rightarrow \omega^* = \langle \omega_1, \omega_2, \dots, \omega_n \rangle = \langle \omega_1, \omega_2, \dots, \langle \bullet \rangle \rangle$.

Definition 10: The colour functions associated with the arcs of the net are the elements of the matrix I^+ (I). They are

defined $\forall \omega^* \in \Omega^*$ and they are built from the following basic (standard) functions or their linear combination [16,17]:

- Projection function (Proj), which selects a component ω_j (colour) of an item ω^* ;
- Identity function (Id), which selects all the components of an item ω^* ;
- Successor (Succ) / Predecessor (Pred) function, which selects some successor or predecessor of a component of an item. For example, the 3rd successor of the k^{th} colour of an item ω^* is defined as follows:

$$\text{Succ}_{k3}: \Omega^* \rightarrow \Omega^*: \langle \omega = (\omega_1, \omega_2, \dots, \omega_k, \dots, \omega_n) \rangle \rightarrow \langle \omega_1, \omega_2, \dots, \omega_k \oplus 3, \dots, \omega_n \rangle, \forall k \in \Omega_k;$$

- Decoloured function (Abs), which transforms a coloured marking in the uncoloured token $\langle \bullet \rangle$.

Remark: It is possible to build a composition of standard colour functions. For example, if $\text{Succ}_{k3}: \Omega^* \rightarrow \Omega^*$ and $\text{Proj}_k: \Omega^* \rightarrow \Omega_k$ then $\text{Proj}_k \circ \text{Succ}_{k3}: \Omega^* \rightarrow \Omega_k$.

C. Dynamic Behaviour

The net evolution, i.e., dynamic behaviour, is supported by the firing of the transitions, which is characterized by the movement of tokens between places. At least two aspects are represented by the evolutions (when a transition fires and the CPN evolves from one state to another one): the first one expresses serial and concurrence events that can be observed, i.e., the interaction states achieved by the agents. The second aspect describes the causally precedence that exists among production components, i.e., physical agents, and interaction acts occurred during production control and management.

Definition 11(Firing Rule): If a transition or firing-mode is effectively enabled, then it can be fired.

Definition 12 (Evolution Rule): The firing of a transition or a firing mode t_i issues a change of the state of the net (change of the marking), which can be represented as follows:

$$M_0 [t_i > M_f$$

being M_0 and M_f the initial and final markings (states).

The evolution of the net is characterized by the token-game, i.e., flow of marks between places.

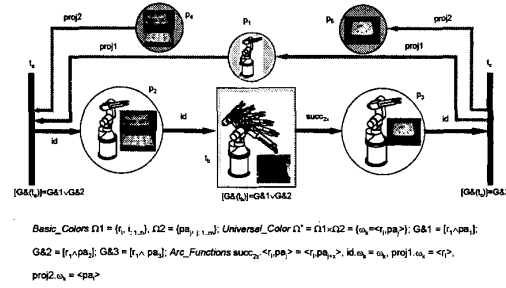
Definition 13 (Sequence of firing): A firing sequence from a marking M_0 is a (possibly empty) sequence of transition sets

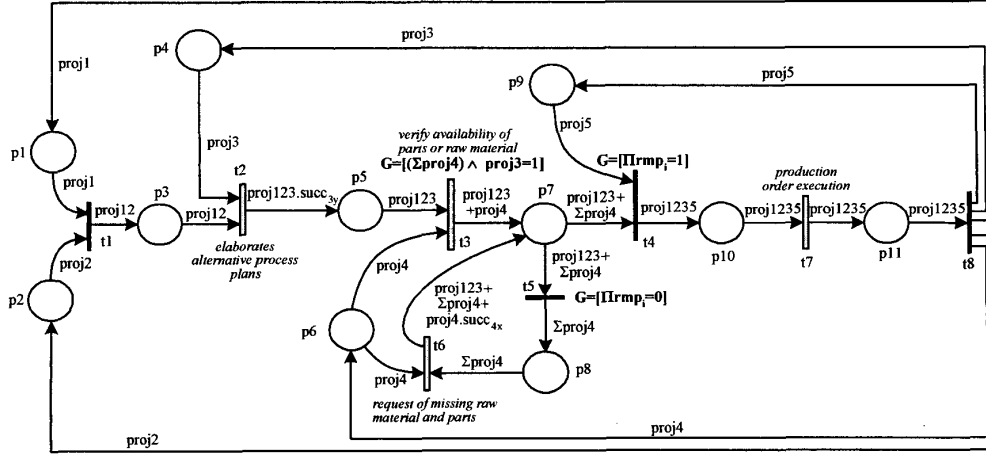
$$\sigma = t_1 t_2 \dots t_k \text{ so that } M_0 [t_1 > M_1 [t_2 > M_2 \dots [t_k > M_k$$

where $M_0 [\sigma >$ denotes that the sequence σ may be fired at M_0 , and $M_0 [\sigma > M_k$ denotes that the firing of σ yields M_k .

D. Graphic Representation

As example, consider Fig. 3 that depicts the graphical and data structure of a sample CPN tailored for holonic control system specifications.





$$M_0(p1) = \{\sum_{[i:1-n]} ph_i\}; M_0(p2) = \{\sum_{[i:1-m]} or_i\}; M_0(p4) = \{\sum_{[i:1-l]} wp_i\}; M_0(p6) = \{\kappa \cdot \sum_{[i:1-q]} rmp_i\}, \kappa \in \mathbb{N}; M_0(p9) = \{\sum_{[i:1-p]} th_i\}$$

Fig. 4 Product Holon CPN Model

IV. MODELLING THE BEHAVIOUR OF THE ADACOR PRODUCT HOLONS USING HIGH LEVEL PETRI NETS

In this section, one of the ADACOR holon classes, namely the product holon, will be modelled using High-Level Petri Nets, illustrated in Fig. 4, with the formal notation, previously described. The number of product holons present in the holonic manufacturing control application is dependent of the manufacturing contextualization, but one and only one model of a product holon is needed. This complex model is capable to represent all products and their instances.

A. Colours Definition

The initial phase in the modelling of product holon class, using High-Level PN, is the definition of the colours related to the product behaviour. During the holon life-cycle, these colours will be managed by the functions associated with the structure of the holons present in the holonic control system.

According the Definition 7, the basic colours defined in this work for the product holon model are:

- **PH** = $\{ph_1, ph_2, \dots, ph_n\}$, which are all possible product holons in the system. In case of 1000000 product holons present in the system, the n value is related to 1000000;
- **Or** = $\{or_1, or_2, \dots, or_m\}$, which are all possible production orders in the system;
- **WP** = $\{wp_1, wp_2, \dots, wp_l\}$, which are all possible work plans in the system;

– **RMP** = $\{rmp_1, rmp_2, \dots, rmp_q\}$, which are all possible raw materials and parts in the system;

– **TH** = $\{th_1, th_2, \dots, th_p\}$, which are all possible task holons in the system.

Some complex colours, according Definition 8, are also defined from the product of two or more basic colour domains:

- $\Omega_{12} = PH \times Or = \langle ph_i, or_j \rangle$, which represents a production order or_j for the product represented by the product holon ph_i (since each product holon can have several instances running at the factory plant).
- $\Omega_{123} = PH \times Or \times WP = \langle \omega_{ak} = \langle ph_i, or_j, wp_z \rangle \rangle$,
- $\Omega_{1234} = PH \times Or \times WP \times RMP = \langle ph_i, or_j, wp_z, rmp_h \rangle$,
- $\Omega_{12345} = PH \times Or \times WP \times RMP \times TH = \langle ph_i, or_j, wp_z, rmp_h, th_r \rangle$, which is the universal colour in the model, i.e. the Cartesian product of all basic colour domains.

One possible colour in real time associated with a real software/hardware component is presented in Fig. 5.

B. Functions Definition

In a CPN model, each input and output arc have associated a function that manipulates the colours components, by removing appropriated components from the input elements and putting appropriated components in the output elements.

In the product holon model, the following functions were defined:

- **proj1**. $\omega_k = \langle ph_i \rangle$, is the projection function that filters the ph_i colour.
- **proj2**. $\omega_k = \langle or_j \rangle$, is the projection function that filters the or_j colour, creating a new instance of a production order.
- **proj3**. $\omega_k = \langle wp_z \rangle$, is the projection function that filters the wp_z colour.
- **proj4**. $\omega_k = \langle rmp_h \rangle$, is the projection function that filters the rmp_h colour.
- **proj5**. $\omega_k = \langle th_r \rangle$, is the projection function that filters the th_r colour.
- **proj12**. $\omega_k = \langle ph_i, or_j \rangle$, filters composed sub-tuples in order to get refined information, in this case the pair product holon (i) and production order (j).
- **proj123**. $\omega_k = \langle ph_i, or_j, wp_z \rangle$, filters ω_k , returning the pair product holon (i), production order (j) and work plan (z).
- **proj1235**. $\omega_k = \langle ph_i, or_j, wp_z, th_r \rangle$, filters ω_k , returning the pair product holon (i), production order (j), work plan (z) and task holon (r).
- **succ_{3y}**. $\omega_k = \langle ph_i, or_j, wp_{z+y} \rangle$, wp_i being one possible Work Plan of the set WP, the function *succ_{3y}* selects the "y" successor of the wp_i , according to *proj1*, *proj2* and *proj3* functions, i.e. that wp that matches with the current product holon-production order pair.
- **succ_{4x}**. $\langle rmp_h \rangle = \langle rmp_{h+x} \rangle$, returns the x successor of the rmp component, i.e. the next rmp component in the product structure.
- **id**. $\omega_k = \omega_k$, which is the identity function, i.e. this function applied to a specific colour gives as result the same colour.

A transition that represents the selection of two distinct actions or the result of a decision for a component has so many firing modes as holons of these types were defined. Only by filtering it will be possible to know which action will be performed or which component is taking the action. As example, in the model to know if all raw material and parts, necessary to execute the product, are available, it is used a guard function that selects the flow of the tokens. The guard $G=[\prod rmp_i=1]$ has the True Boolean value if all instances of RMP colour domain are True, i.e. all raw material or parts are available to produce a product. Otherwise, this guard function is false, but the guard $G=[\prod rmp_i=0]$ has the True Boolean value, re-directing the flow of the tokens to the need to request the missing raw materials or parts.

Analysing in more detail the product holon model, it is possible to verify that it contains recursivity. In the case that a

sub-product is requested through the "request of missing raw material and parts" activity, this model is recursively invoked, implying a production order (belongs to Or), and the sub-product will be considered by a product holon (belongs to PH).

V. FORMAL VALIDATION OF THE MODEL

Taking into consideration the exemplary marking presented in Fig. 5, which represents a state of the net depicted in Fig. 4, the following are some of the specifications that can be validated:

- There are two products with their corresponding production orders that are being currently processed:
 - The product holon ph_i , which is in charge of the production order or_j , has already assigned/elaborated a work plan (wp_k) to the product and is ready to start the verification of the availability of parts and raw material that are necessary for the production process.
 - The product holon ph_3 , which is in charge of the production order or_j , has already finished the verification of available parts and raw material and it has also synchronised its actions with a task holon (th_p). The last is ready to start with the control of the execution of the production order using the work plan wp_s .
- The system is ready to initiate the process of (n-2) new products, to process (m-2) new orders, to assign (l-2) work plans, and to synchronize the activities with (q-1) task holons.

Example of Marking	
P1	$\{\langle ph_1 \rangle, \langle ph_2 \rangle, \dots, \langle ph_4 \rangle, \langle ph_{(i-1)} \rangle, \langle ph_{(i+1)} \rangle, \dots, \langle ph_n \rangle\}$
P2	$\{\langle or_1 \rangle, \dots, \langle or_3 \rangle, \dots, \langle or_{(j-1)} \rangle, \langle or_{(j+1)} \rangle, \dots, \langle or_m \rangle\}$
P3	
P4	$\{(\sum_{i: 1-l} wpi) - wpk - wps\}$
P5	$\{\langle ph_i, or_2, wp_k \rangle\}$
P6	$\kappa. \{(\sum_{i: 1-l} rmp_i)\}$
P7	
P8	
P9	$\{(\sum_{i: 1-l} th_i) - th_q\}$
P10	$\{\langle ph_3, or_j, wp_s, th_p \rangle\}$
P11	

Fig. 5 Example of Marking

The characteristics addressed above and other structural and behavioural specifications of the modelled system can be formal validated using the information contained in the incidence matrix of Fig. 6.

I	t1	t2	t3	t4	t5	t6	t7	t8
p1	-proj1							proj1
p2	-proj2							proj2
p3	proj12	-proj12						
p4		-proj3						proj3
p5		proj123 succ ₃	-proj123					
p6			-Σ proj4					Σ proj4
p7			proj123+ Σ proj4	-proj123+ Σ proj4	(proj123+ Σ proj4)	proj123+ Σ proj4+ proj4.succ ₄		
p8					Σ proj4	-Σ proj4		
p9				-proj5				proj5
p10				proj1235			-proj1235	
p11							proj1235	-proj1235

Fig. 6 Incidence Matrix

Applying well-known concepts of the “functional analysis” and the “linear algebra”, the matrix of Fig. 6 will be diagonalized. The procedure that conduces to the diagonalisation of the Incidence matrix allows obtaining the set of right- and left-annulators, which are related to sub-sets of places and sub-sets of transitions (and their firing modes) of the Coloured Petri Net model. In the literature, these sub-sets are called place- and transition-flows.

Remark: The calculus of place- and transition-flows of a CPN is out of the scope of this paper. For a better understand of the subject, the authors recommend consulting the references [12,17,18] and the references therein.

	Place-flow 1	Place-flow 2
P1	1	0
P2	0	0
P3	proj1	0
P4	0	0
P5	proj1	0
P6	0	0
P7	proj1	0
P8	proj1	0
P9	0	1
P10	proj1	proj5
P11	proj1	proj5

Fig. 7 Place-flows of Incidence Matrix

As an example, two place-flows of the Incidence matrix are shown in Fig. 7. These sets of place-flows generate a set of place-invariants (PI). For example:

PI1:

$$M(p1) + \langle \text{proj1} \rangle \cdot [M(p3) + M(p5) + M(p7) + M(p8) + M(p10) + M(p11)] = M_0(p1) + \langle \text{proj1} \rangle \cdot [M_0(p3) + M_0(p5) + M_0(p7) + M_0(p8) + M_0(p10) + M_0(p11)] = \sum_{i=1..n1} \langle \text{ph}_i \rangle$$

PI2:

$$M(p9) + \langle \text{proj5} \rangle \cdot [M(p10) + M(p11)] = M_0(p9) + \langle \text{proj5} \rangle \cdot [M_0(p10) + M_0(p11)] = \sum_{i=1..n2} \langle \text{th}_i \rangle$$

The processing of these invariants allows the validation of different specifications, for example:

Proposition 1: At any one time, the maximum number of product holons that can be set into operation by ADACOR is limited to n , i.e., the initial marking of place p1, of the net depicted in Fig. 4.

Proof: From PI1, if this is multiplied by $\langle \text{ph}_i \rangle$, then

$$\forall \omega = \langle \text{ph}_i, \text{or}_j, \text{wp}_i, \text{rmp}_i, \text{th}_i \rangle \in \Omega^*$$

$$\begin{aligned} \langle \text{ph}_i \rangle \cdot \{M(p1) + \langle \text{proj1} \rangle \cdot [M(p3) + M(p5) + M(p7) + M(p8) + M(p10) + M(p11)]\} &= \\ = \langle \text{ph}_i \rangle \cdot \{M_0(p1) + \langle \text{proj1} \rangle \cdot [M_0(p3) + M_0(p5) + M_0(p7) + M_0(p8) + M_0(p10) + M_0(p11)]\} &= \langle \text{ph}_i \rangle \cdot \sum_{i=1..n1} \langle \text{ph}_i \rangle = n \end{aligned}$$

With this equality it is possible to conclude that,

$$\langle \text{ph}_i \rangle \cdot \langle \text{proj1} \rangle \cdot [M(p3) + M(p5) + M(p7) + M(p8) + M(p10) + M(p11)] \leq n, \text{ i.e. the number of product holons in activity,}$$

and

$$\langle \text{ph}_i \rangle \cdot M(p1) \leq n, \text{ i.e. the number of product holons idle.}$$

The proof is straightforwardly obtained from both inequalities.

Corollary1: For the exemplary state of the model presented in Fig. 3, the application of the results of proposition 1, when the product holon $\langle \text{ph}_3 \rangle$ is assigned, renders the following main conclusion:

$$\begin{aligned} \langle \text{ph}_3 \rangle \cdot \langle \text{proj1} \rangle \cdot [M(p3) + M(p5) + M(p7) + M(p8) + M(p10) + M(p11)] &= \\ = \langle \text{ph}_3 \rangle \cdot \langle \text{proj1} \rangle \cdot [0 + \langle \text{ph}_i, \text{or}_2, \text{wpk} \rangle + \kappa \cdot \{\sum_{i=1..n1} \text{rmp}_i\}] + 0 + 0 + \{\sum_{i=1..n1} \text{th}_i\} - \{\sum_{j=1..n2} \text{th}_j\} + \{\langle \text{ph}_3, \text{or}_j, \text{wp}_s, \text{th}_p \rangle\} &= \\ = \langle \text{ph}_3 \rangle \cdot [0 + 0 + 0 + 0 + \langle \text{ph}_3 \rangle + 0] = 1 \end{aligned}$$

and

$$\langle \text{ph}_3 \rangle \cdot M(p1) = \langle \text{ph}_3 \rangle \cdot \{\langle \text{ph}_1 \rangle, \langle \text{ph}_2 \rangle, \dots, \text{ph}_4, \langle \text{ph}_{(i-1)} \rangle, \langle \text{ph}_{(i+1)} \rangle, \dots, \langle \text{ph}_n \rangle\} = 0$$

This means, that at any one time, a product holon can only be assigned to the process of a product.

Corollary 2: It is possible to formal validate that a product holon can be assigned to produce at any time different orders). Since the colour sets defined in Section IV-A are Bags, a colour tone can be present with a given multiplicity, i.e., $u \cdot \langle \text{ph}_i \rangle$. This means that additionally, a product holon can at any time handle the execution of several different orders, in the maximum of m (cardinality of the set Or). For example, an order or(2) to produce the product A with due date D1, ..., and an order or(12) to produce the product A with due date D4.

Remarks: The ADACOR architecture (see [7]) was developed following the specification that a task holon needs of the following information/elements to supervise the execution of a production order: information about the production order (quantity, due date,...), the physical availability of raw material or parts that belongs to the product BOM, and a set of process plans that indicates how to produce the product. At the end of execution, the task holon provides to the product holon some information related to how the production order was really executed, such as start and end dates of each

operation, the indication about the resources that executed the operations, etc.

Proposition 2: At any one time, the maximum number of task holons that can be set into operation by ADACOR is limited to p , i.e., the initial marking of place p_9 , of the net depicted in Fig. 4. This value is equal to the sum of all production orders launched to the shop floor to be executed (those in places p_{10} and p_{11}), plus the set of task holons that are already idle.

Proof: Immediate from the structure of place-invariant relationship PI_2 .

Corollary: The transition t_4 models activities related to the decision if all raw materials and parts are physically available to produce a product. The transition t_7 models the production order execution. This transition initially launches the task holon (or starts or gives the indication to the task holon), passing the production order information and a set of process plans. After that, the control is passed to the task holon (firing colour of the transition t_7), which is responsible for the execution of the production order. When the production order is finished, the task holon returns the control to the product holon passing back information about the order execution, by means of the firing of transition t_8 . The mutual exclusion relationship between the markings of p_9 , p_{10} and p_{11} , is closed related to the firing sequence associated to the set of transitions addressed above.

VI. CONCLUSIONS AND FUTURE WORK

The formal specification of holonic control systems assumes a crucial role to understand and synthesize the behaviour of complex and concurrent systems, such as the flexible manufacturing systems.

A brief overview of the latest works published in the area (see e.g. [19, 20] and our last reports [8] and [11]) allowed us to identify a set of weak points in using the PN-formalism and other similar extensions of this tool. This is particularly true if the system presents many instances of the same component (e.g., n resources need n operational holons). In this case, the model will grow (structure and components) in a non-controllable manner. In our opinion, the use of High-Level PN, allows to reduce this complexity, by compressing the representation of states, actions and events, to overcome the identified limitations and to support more complex and bigger coordination scenarios, like these presented here.

In this paper a High-Level PN formalism, i.e. Coloured Petri Nets, is used to model the dynamic behaviour of ADACOR product holons and formally validated this model allowing to verify and optimise the functioning of the system.

Applying the same methodology, the set of specifications of the behaviour of each ADACOR-holon and of the whole ADACOR-control system can be validated, and of course optimised.

ACKNOWLEDGEMENTS

The first author would like to acknowledge the PRODEP program for the financial support granted (PRODEP III-5.3/2000).

REFERENCES

- [1] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts and P. Peeters, "Reference Architecture for Holonic Manufacturing Systems: PROSA", *Computers In Industry*, vol. 37, 1998, pp. 255-274.
- [2] K. Fisher, "Agent-Based Design of Holonic Manufacturing Systems", *Journal of Robotics and Autonomous Systems*, Elsevier Science B.V., vol.27, pp 3-13.
- [3] M. Fletcher, V. Marik and P. Vrba, "Design Issues in Holonic Inventory Management and Material Handling Systems", in *Knowledge and Technology Integration in Production and Services: Balancing Knowledge and Technology in Production and Service Life Cycle*, V. Marik, L. M. Camarinha-Matos and H. Afsarmaneh (eds.), Kluwer Academic Publishers, 2002, pp. 271-280.
- [4] L. Gou, P. B. Luh and Y. Kyoya, "Holonic manufacturing scheduling: architecture, cooperation mechanism, and implementation", *Computers in Industry*, vol. 37, 1998, pp. 213-231.
- [5] A. W. Colombo, R. Neubert and R. Schoop, "A Solution to Holonic Control Systems", in *Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation*, Sophia/Nice, France, 2001, pp 489-498.
- [6] Journal of Applied Systems Studies, Special Issue on Industrial Applications of Multi-Agent and Holonic Systems, 2(1), 2001.
- [7] P. Leitão and F. Restivo, "Adaptive Production Control Systems", in *Proceedings of special session on Agent-based Intelligent Automation and Holonic Control Systems of the 28th Annual Conference of the IEEE Industrial Electronics Society*, Sevilla, Spain, 5-8 November, 2002, pp 2968-2973.
- [8] P. Leitão, A.W. Colombo and F. Restivo, "An Approach for the Formal Specification of Holonic Control Systems", to appear in the *1st International Conference on Applications of Holonic and Multi-Agent Systems (HoloMAS)*, September 1-3, 2003, Prague, Republic Czech.
- [9] T. Murata, "Petri Nets: Properties, Analysis and Applications", *Proceedings of the IEEE*, 77(4), 1989, pp 541-580.
- [10] A.W. Colombo and R. Carelli, "Petri Nets for Designing Manufacturing Systems", *Computer-Assisted Management and Control of Manufacturing Systems*, S.G. Tzafestas (ed.), cap. 11, Springer-Verlag 1997.
- [11] P. Leitão, A.W. Colombo and F. Restivo, "A Formal Validation Approach for Holonic Control System Specifications", submitted to the *9th IEEE International Conference on Emerging Technologies and Factory Automation*, September 16-19, 2003, Lisboa, Portugal.
- [12] A. W. Colombo, R. Neubert and B. Süßmann, "A Colored Petri Net based Approach Towards a Formal Specification of Agent-Controlled Production Systems", in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Tunisia, 2002.

- [13] K. Jensen, "Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use", vol. 2, 1995, Springer Verlag.
- [14] B. Zhou, L. Wang and D. Norrie, "Design of Distributed real-time control agents for intelligent manufacturing systems", in *Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems*, Leuven, Belgium, 1999, pp. 237-244.
- [15] K. Feldmann and A. W. Colombo, "Material Flow and Control Sequence Specification of Flexible Production Systems using Coloured Petri nets", *International Journal of Advanced Manufacturing Technology (IJAMT)*, vol. 14, 1998, pp. 760-774. Springer Verlag, London.
- [16] M. Couvreur and Martinez, "Linear Invariants in Commutative High-Level Nets", *Advances in Petri Nets, Lecture Notes in Computer Science*, vol. 483, pp. 146-165. Springer Verlag, 1990.
- [17] S. Haddad and J. M. Couvreur, "Towards a General and Powerful Computation of Flows for Parametrized Coloured nets", in *9th European Workshop on Applications and Theory of Petri Nets*, Venice, Italy, 1988.
- [18] K. Jensen, "Coloured Petri nets. Basic Concepts, Analysis methods and Practical Use", vol. 1, Monographs on Theoretical Computer Science, Springer Verlag, 1992.
- [19] V. Vyatkin, H-M. Hanisch and G. Ivanov, "Application of Formal Methods for Deep Testing of Controllers in Holonic Systems", in proceedings of the 1st IEEE International Conference on Information Technology in Mechatronics (ITM2001), Istanbul, Turkey, 1-3 October (2001) 53-58.
- [20] G. Frey, M. Minas and K. John, "Steuerungsentwurf mit Petrinetzen. SPSMagazin", Verlag Marburg, 4/5 (2002) 44-47.